ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν Πανεπιστήμιον Αθηνών
ΙΔΡΥΘΕΝ ΤΟ 1837

Forecasting CMEs using Image Processing & Neural Networks

Savvas Raptis
KTH - Department of Space and Plasma Physics

SpaceCoffee 43
Wednesday 19/12/2018 - 16.00

Preventing a disaster

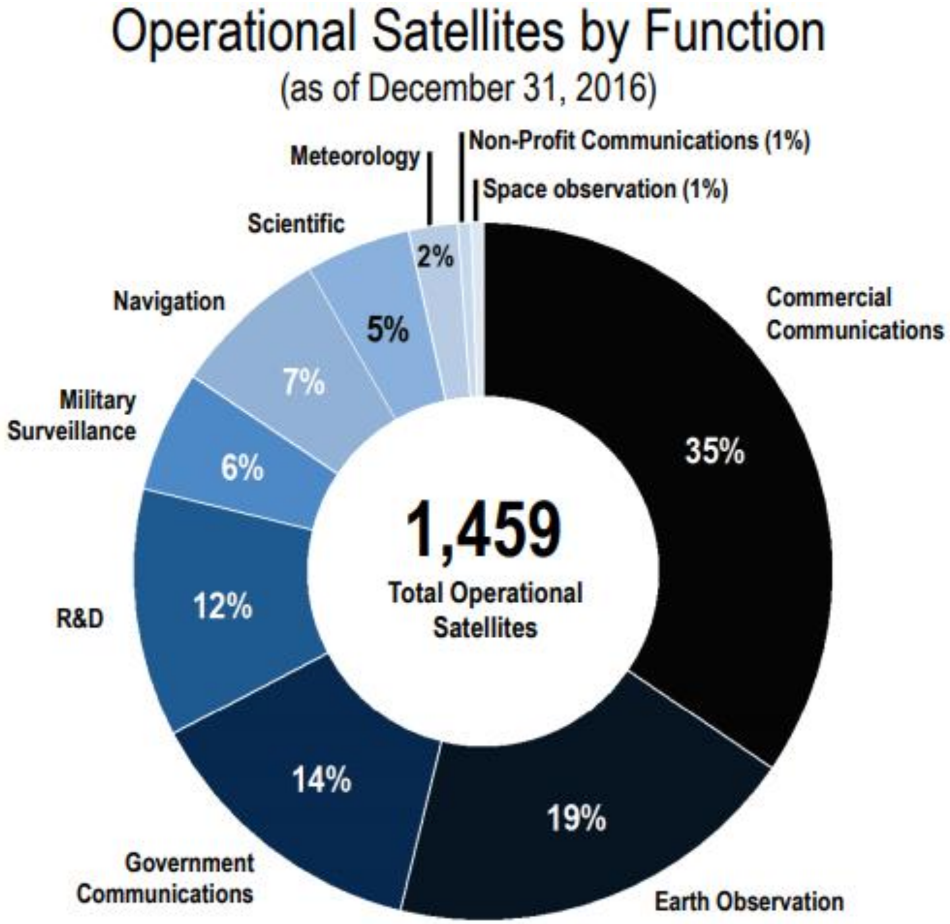


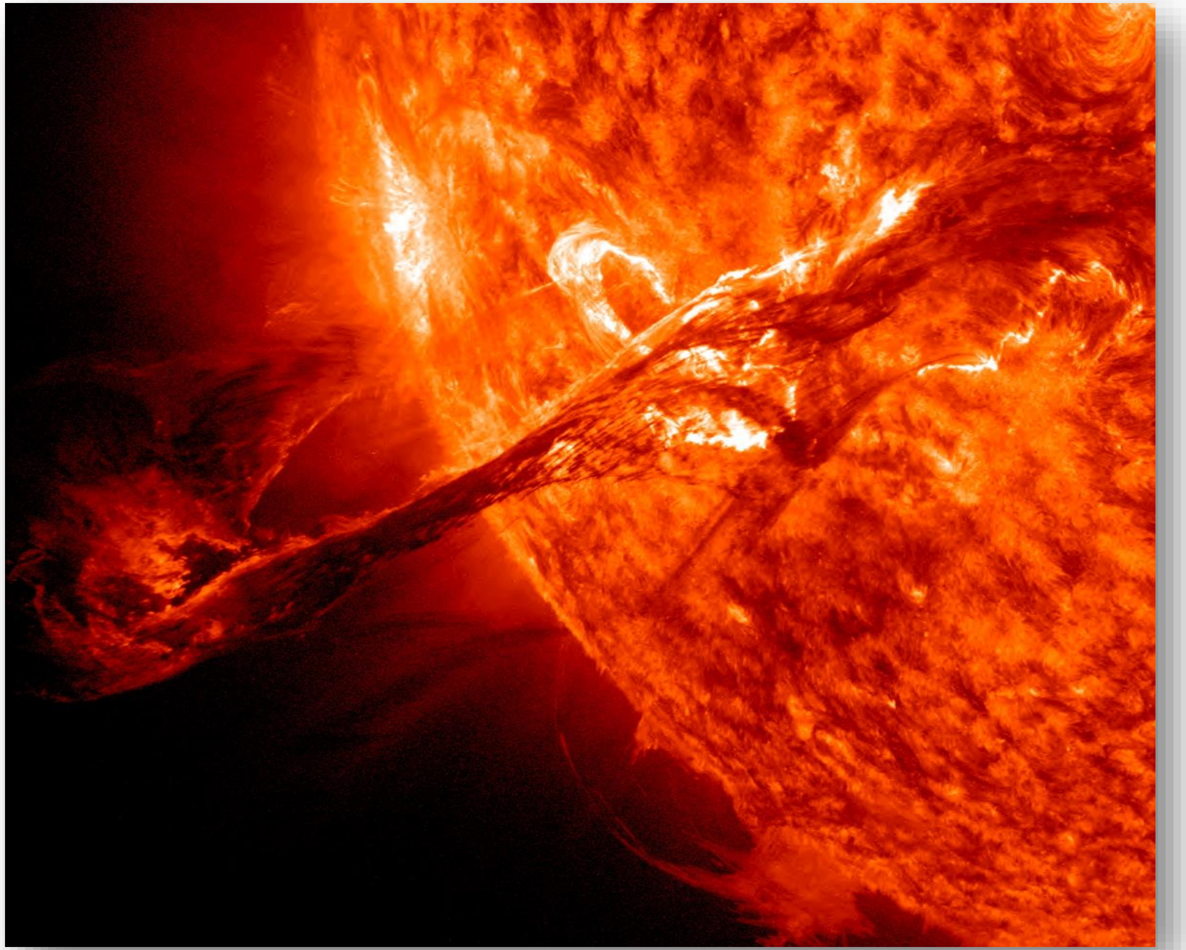
*Figure Courtesy: SIA (Satelite Industry Association)

If Satellites Stop:

- No Telecommunications
- No Military surveillance
- No Weather forecast
- No GPS

What can cause these problems?

Coronal Mass Ejections (CMEs)



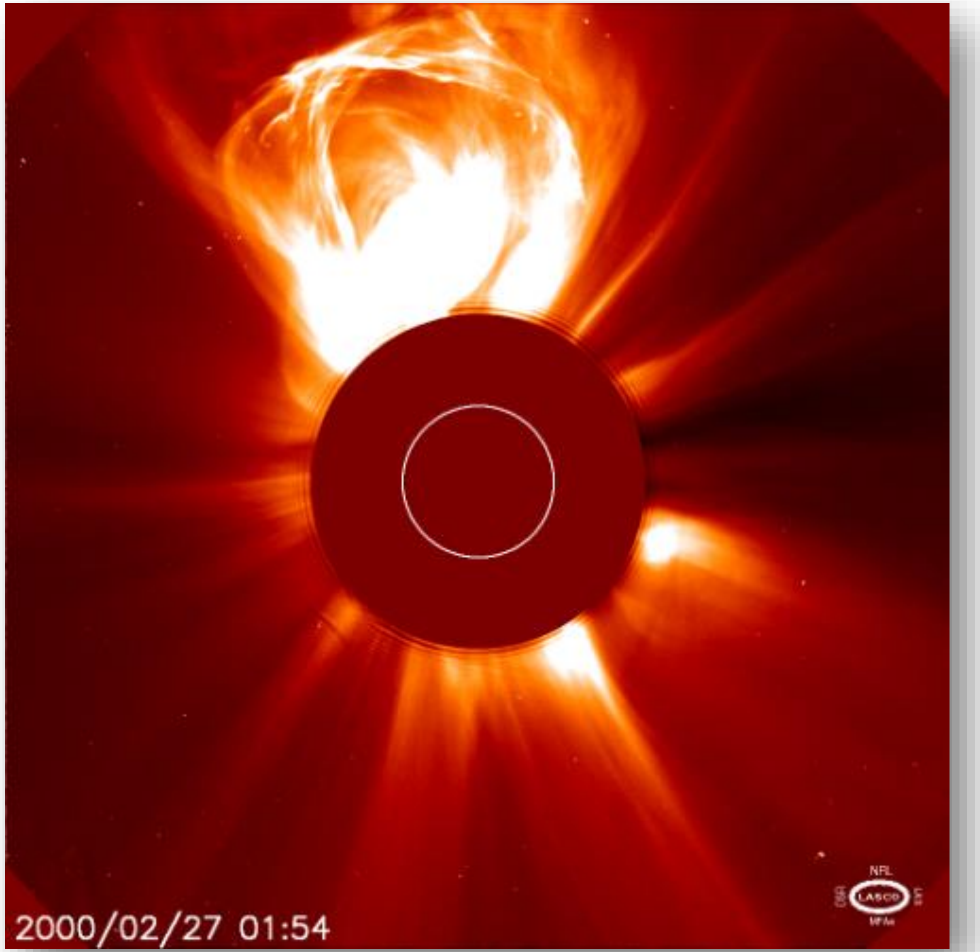
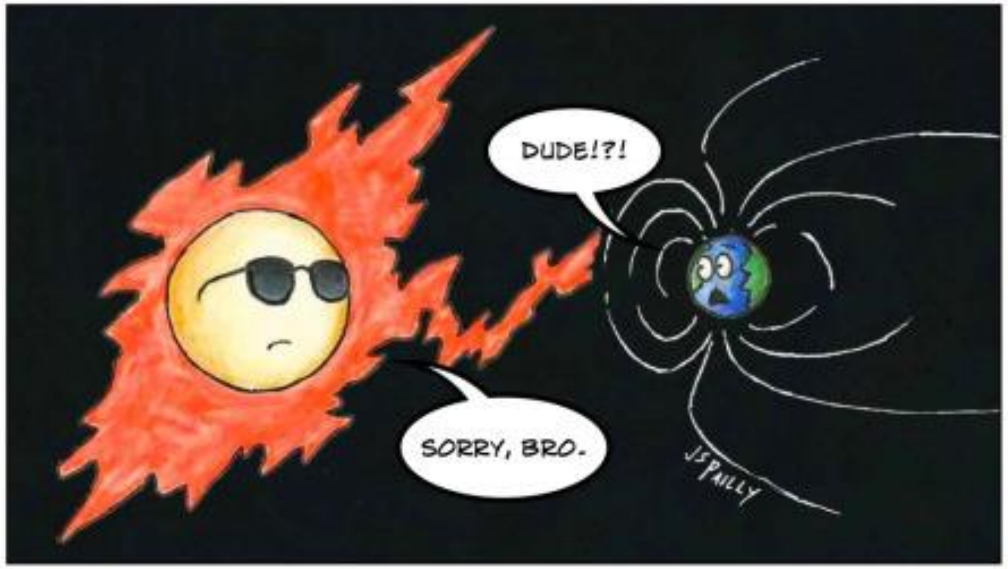


*Figure Courtesy: NASA/ESA, SDO and SOHO satellites



Theory

*Figure Courtesy: https://planetpailly.com/

Coronal Mass Ejections – CMEs

<u>CMEs</u>
Energetic events from the Sun

⇩

Particles in space
+
Disturbances to Earth's magnetic field

⇩

Problems in
Satellites/Communications/Grids/infrastructures etc.

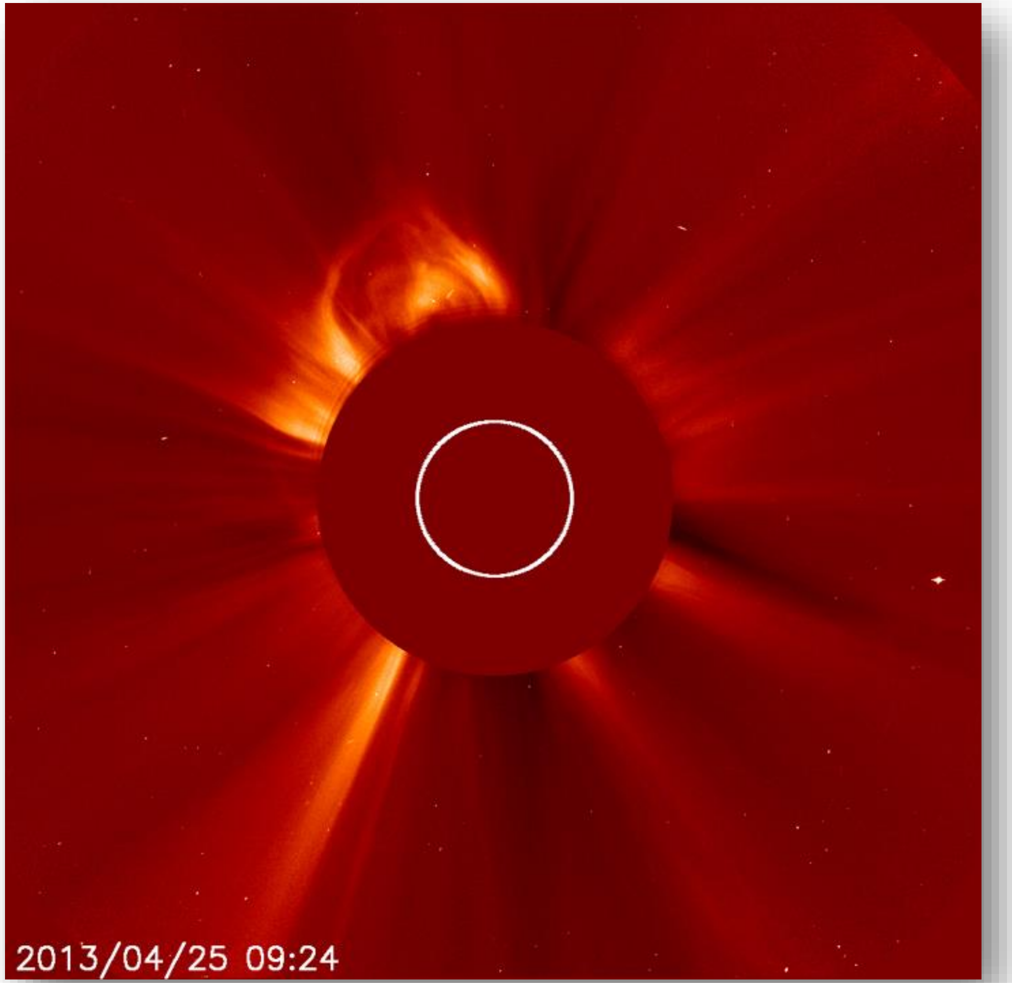


*Figure Courtesy: NASA/ESA, SOHO satellite

Halo CMEs

<u>Halo CMEs</u>
Earth-directed CMEs. Can be seen from coronagraph.

<u>Why important?</u>
Going to Earth

⇩

More effects on mankind

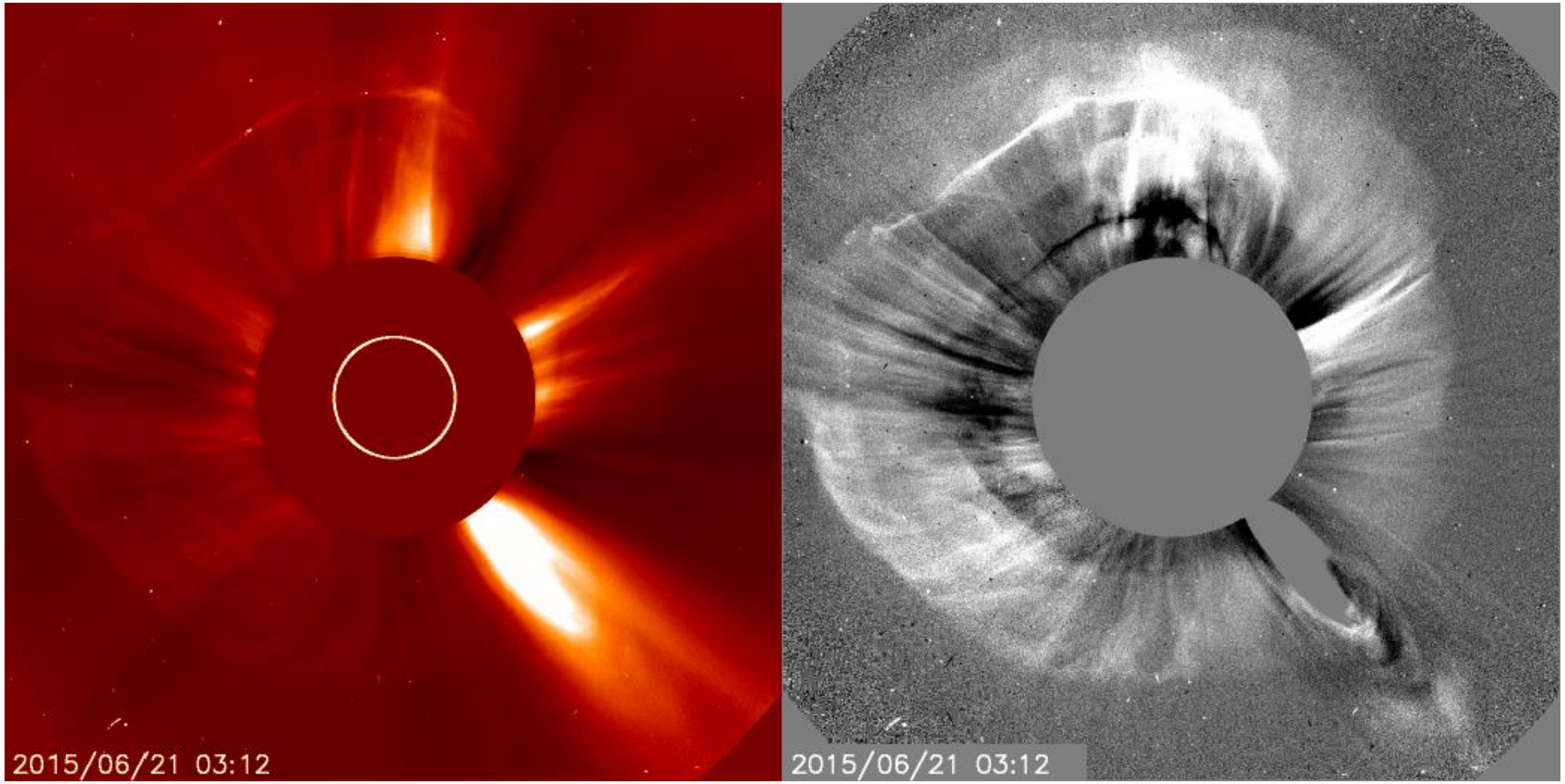


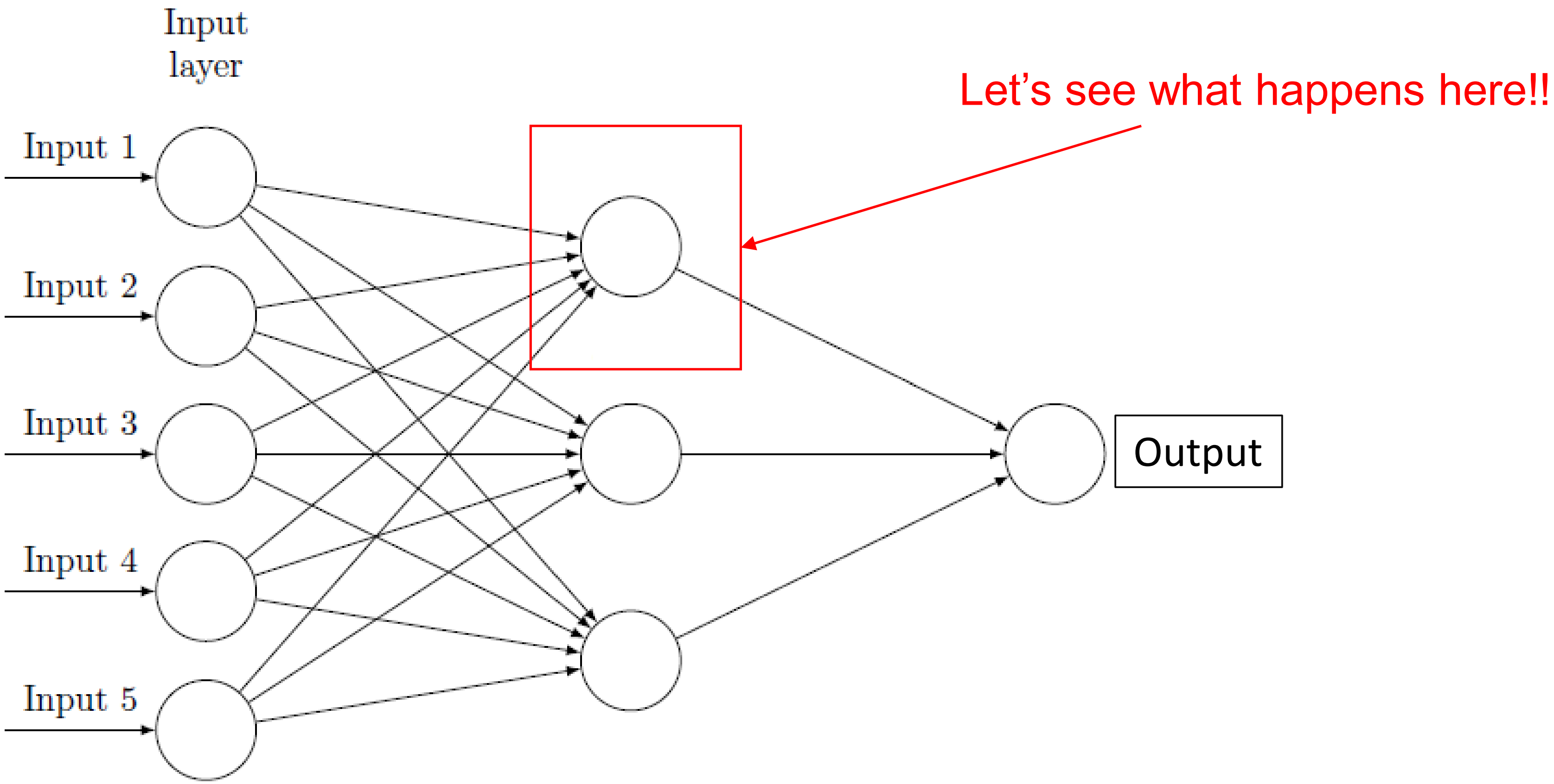
*Figure Courtesy: NASA/ESA, SOHO satellite

What is machine learning & A.I ?

Making the computer "learn" from data without being explicitly programmed

Neural Networks

Neural Networks



A Neural Network Input and Output

$$y = f(x_1 w_1 + x_2 w_2 + \cdots + x_n w_n + b)$$

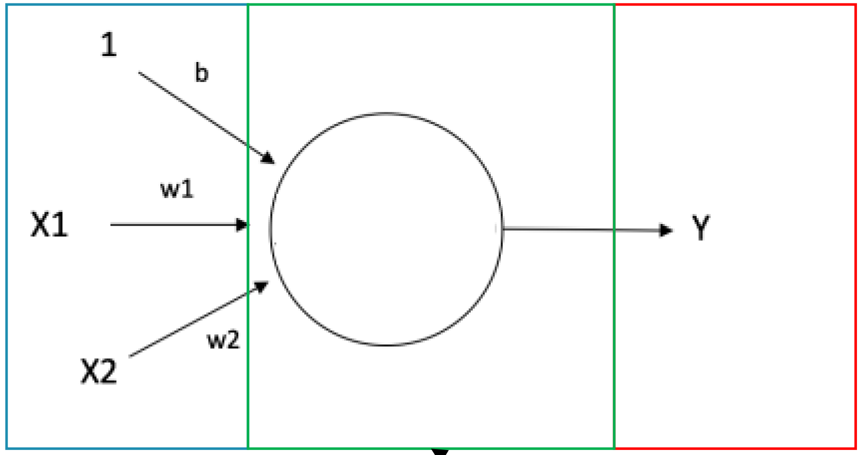
y : Output of Neuron

x_i: Inputs of Neuron

w_i: Weights of each Input

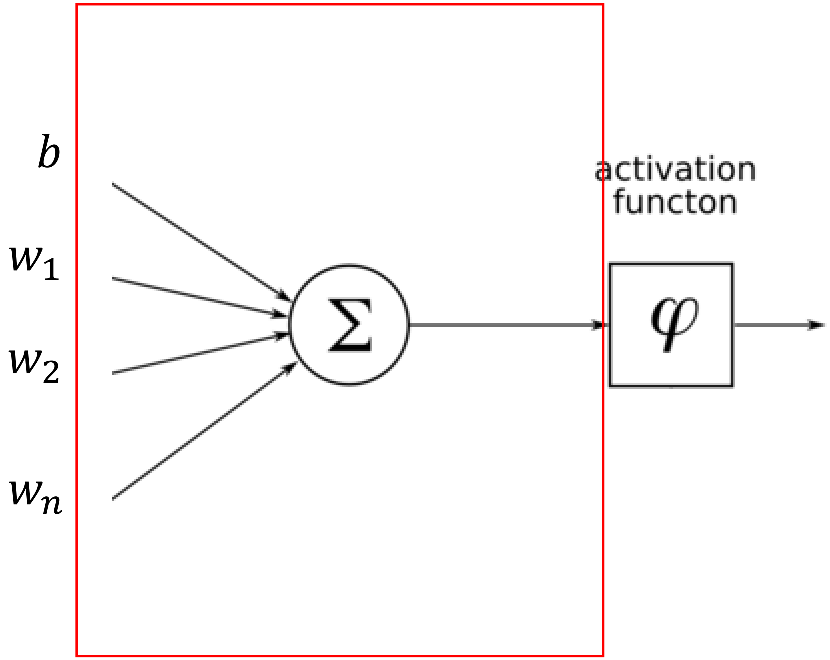
b : bias for each neuron

Random Numbers



Magic happens here

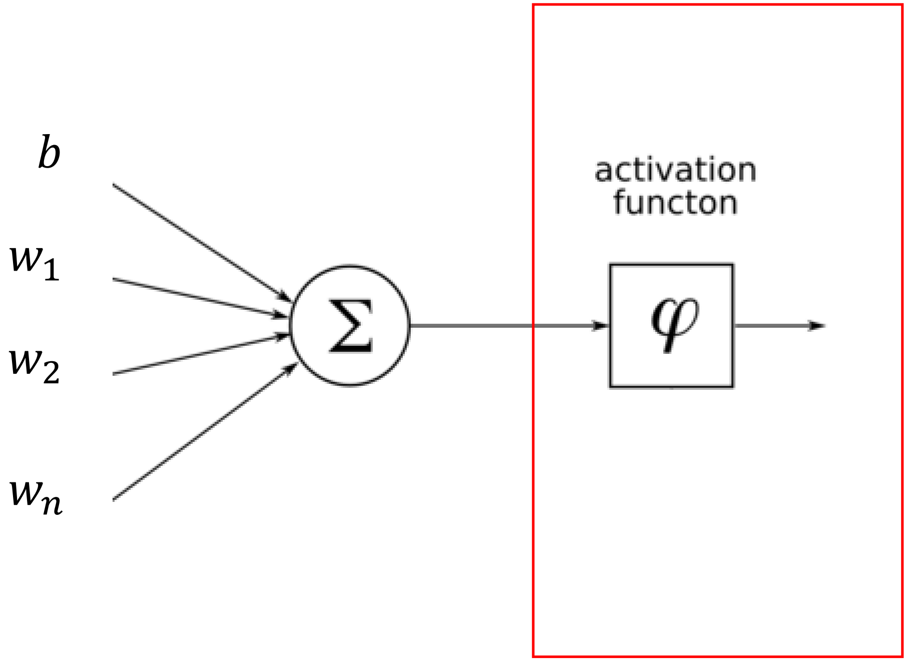
Activation Function



Sum of all Data(x_i), Weights (w_i) and Biases (b)

$$z = \sum x_i w_i + b$$

Activation Function

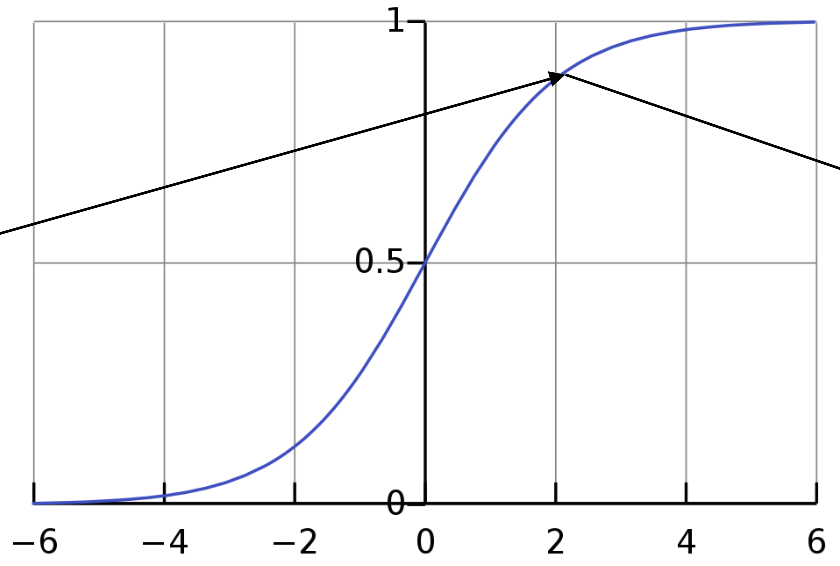


Apply $f(z)$ depending on **goal**

Activation function Examples

Goal: Classification

$\sum x_i w_i + b = 2$

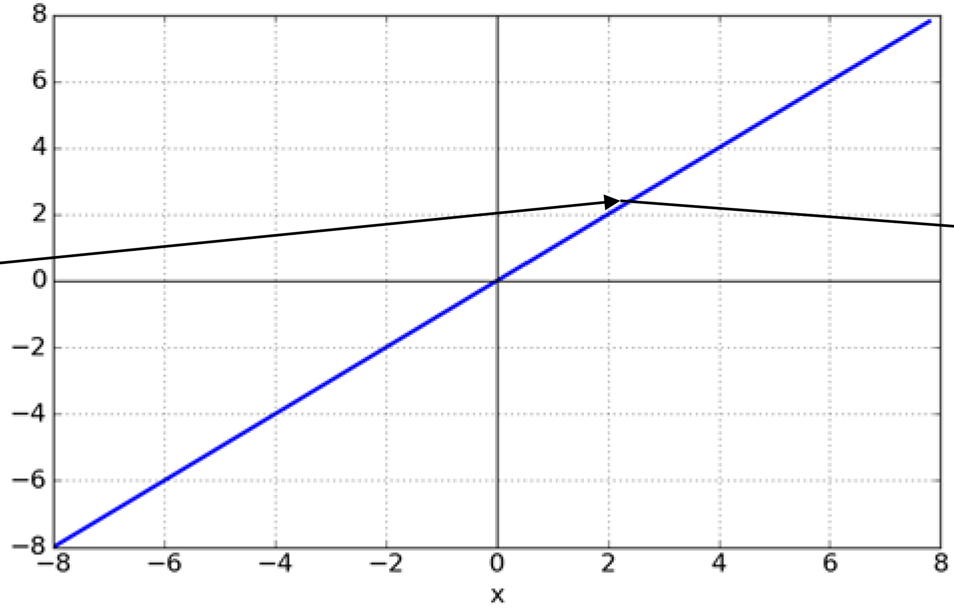


$y = 0.8 \rightarrow Class\ 1$

Activation function Examples

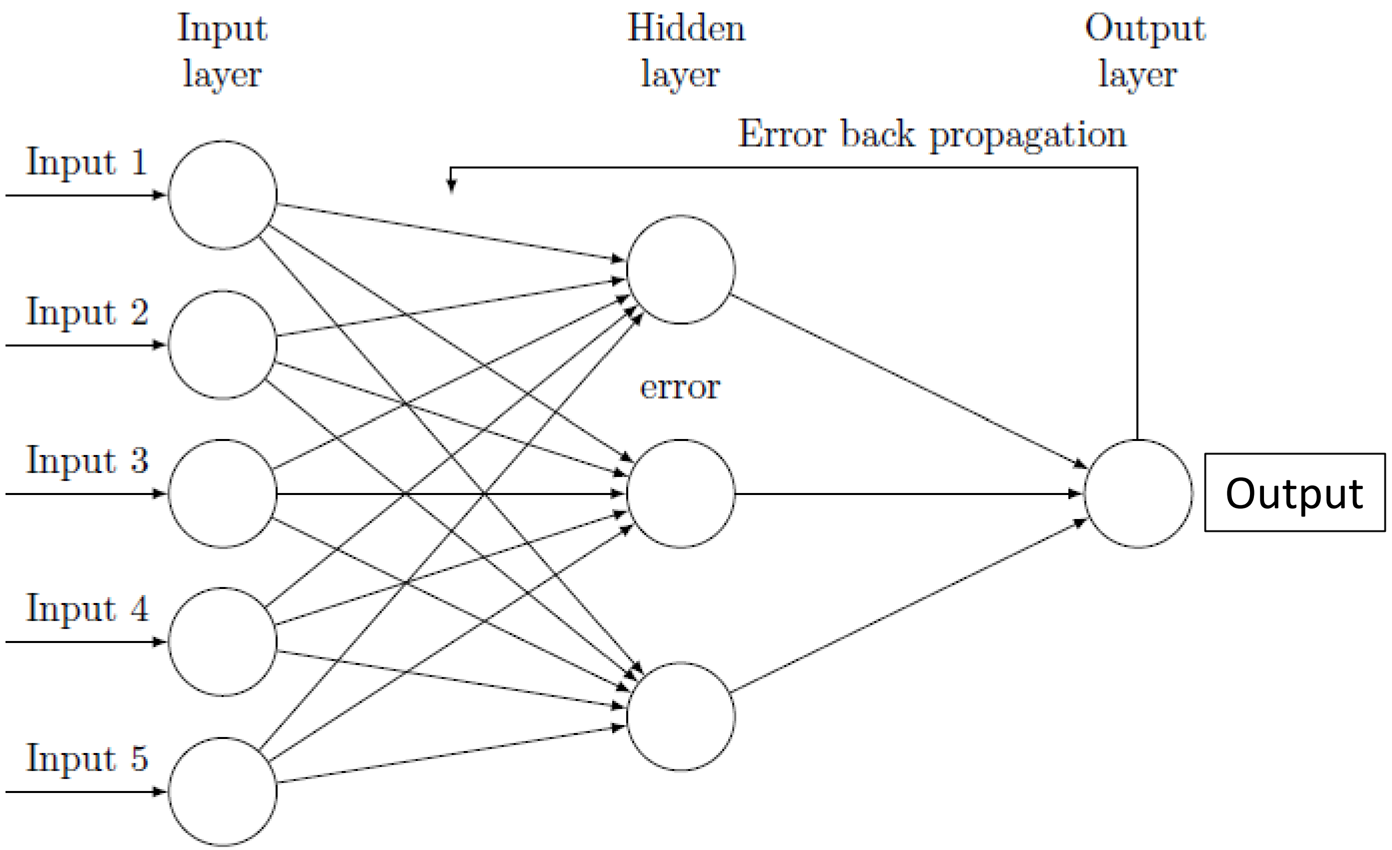
Goal: Regression

$$\sum x_i w_i + b = 2$$

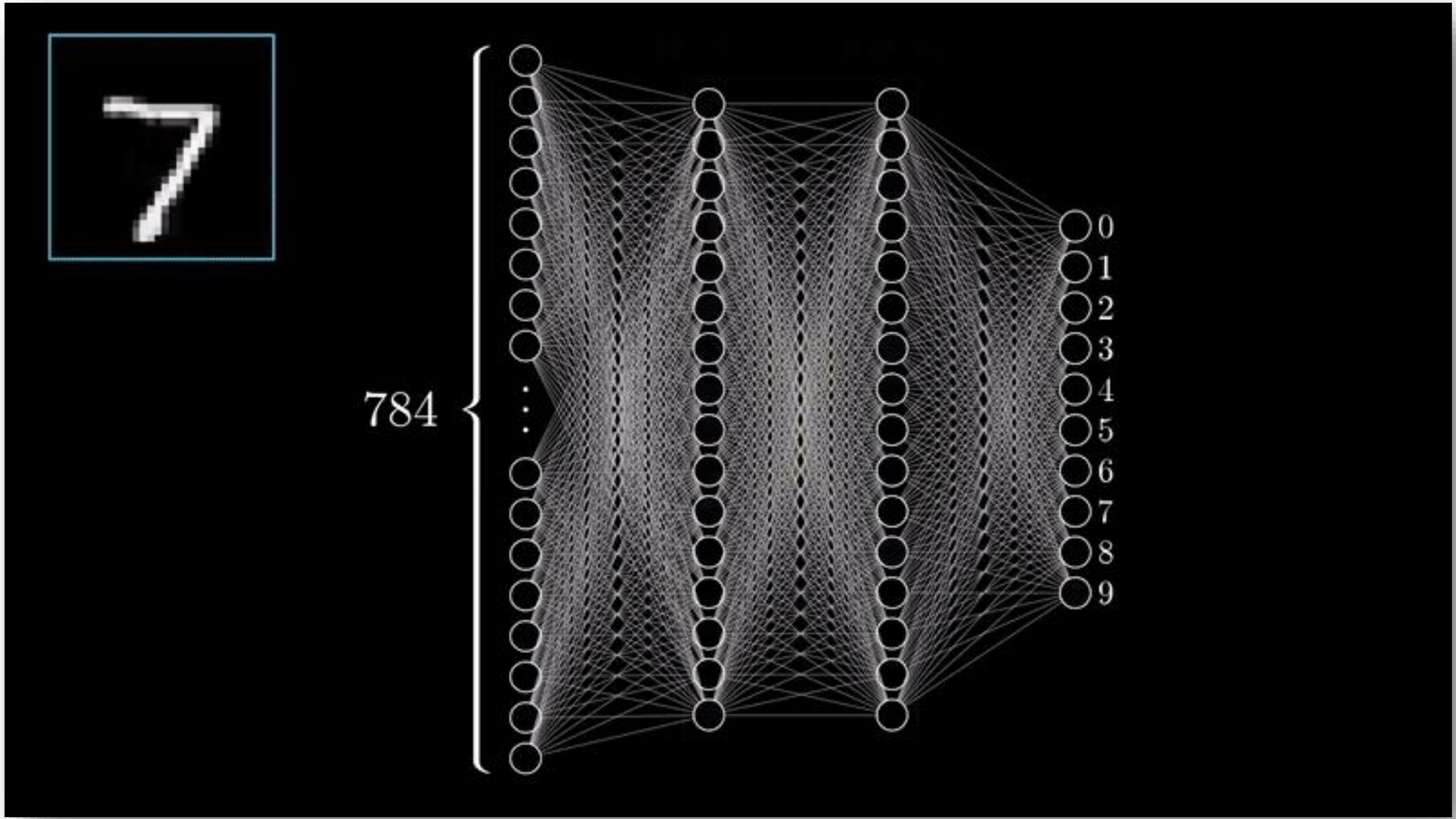


$$y = 2$$

Neural Networks & Backpropagation



A Trained Neural Network

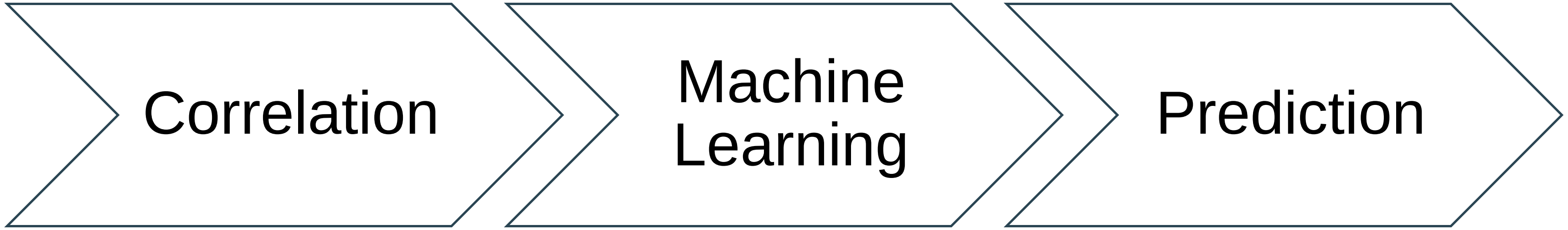


*Video Courtesy**: 3Blue1Brown** (Check him on YouTube!)

Neural Networks in Practice

- Problem: Predict DST Index
- Idea: *Quantities* are **correlated** with *DST* index

(Dynamic Pressure, Magnetic field etc.)



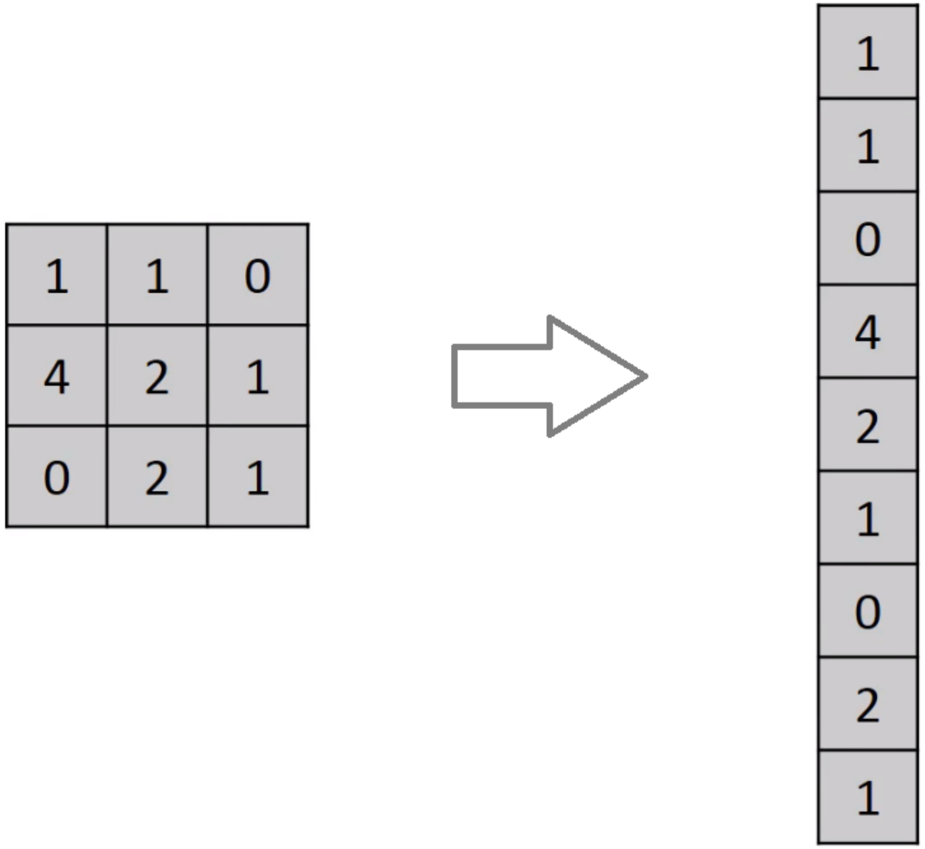
Example: Neural Networks on DST

INPUT: QUANTITIES FROM SATELLITE ACE

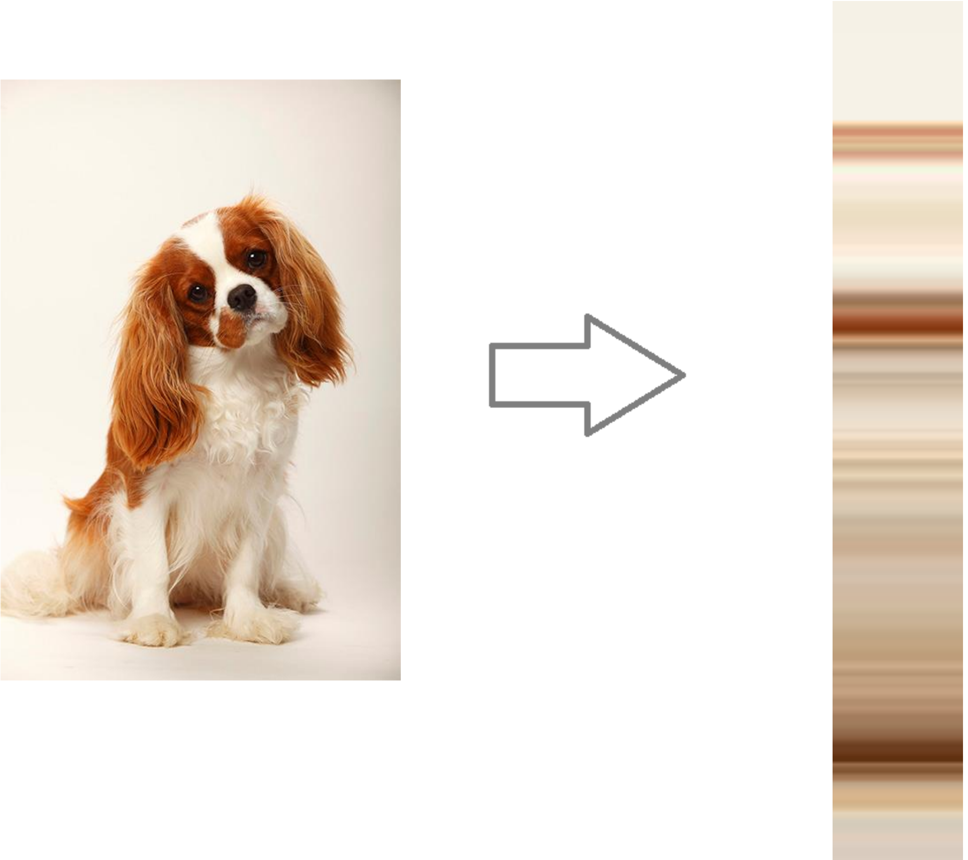
4 LAYERS OF (20, 10, 5, 1) NEURONS
500 EPOCHS

OUTPUT: DST INDEX
(1 HOUR LATER)

Neural Networks with Images



Neural Networks with Images – Dog example



Convolution Neural Networks

Convolution Neural Network (CNN) Layers

<u>Convolution</u>
Extract features & Keep spatial relationship

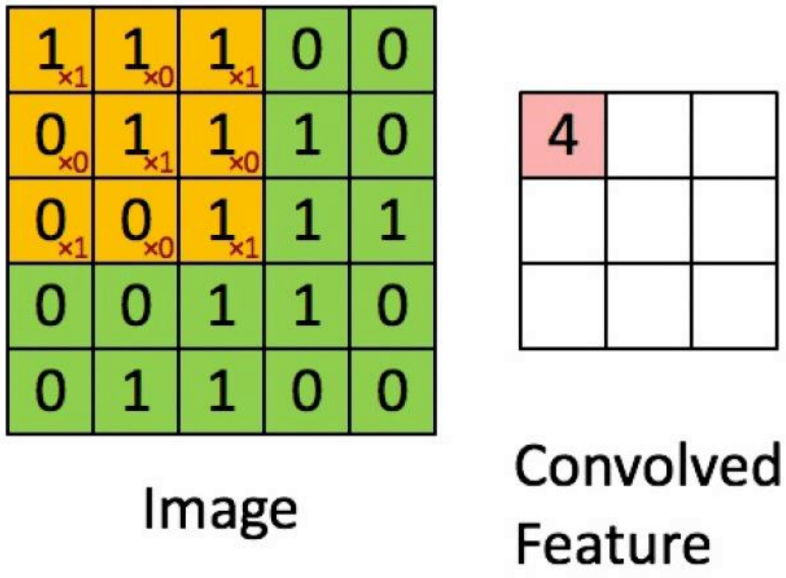


*Figure Courtesy: Erik Reppel

<u>Pooling/Subsampling</u>
Reduce dimensionality & retain information

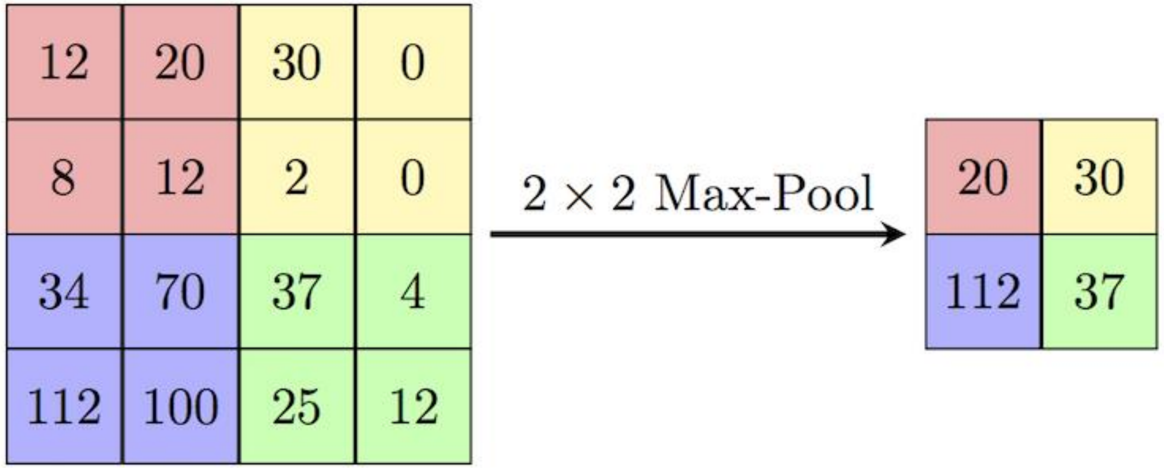


*Figure Courtesy: Cambridge Spark Ltd

Example of CNN

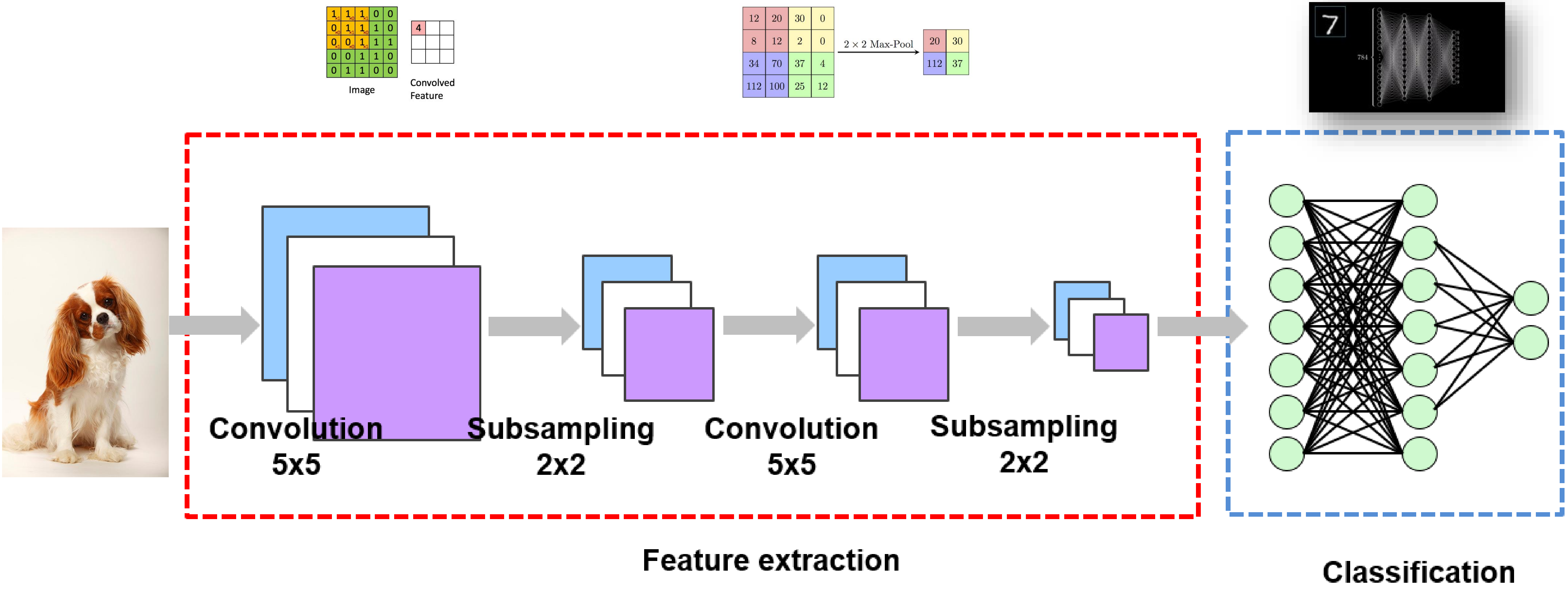
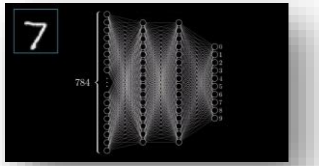


*Figure Courtesy: Suhyun Kim iSystems Design Labs

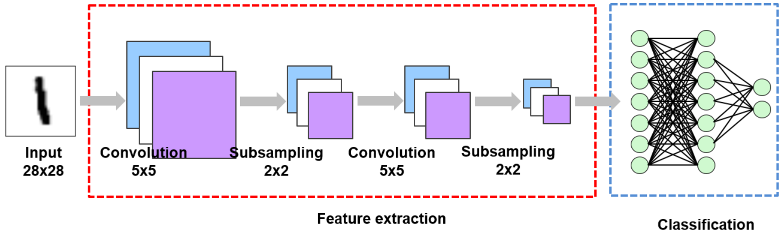
NN vs CNN

Input: MNIST database



Neural Network Result:

97.3%



Convolution Neural Network Result:

99.07%

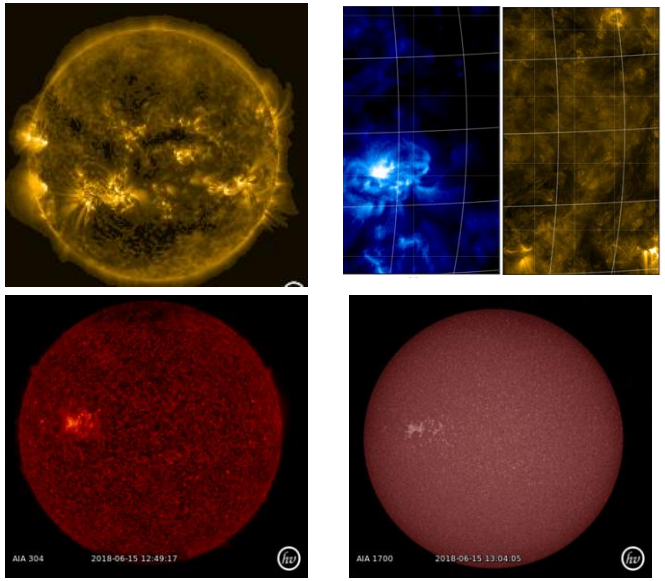
Analysis

Main Goal

Forecast the emerged CMEs using solar images taken from SDO and CNN

Input

SDO Images during 2014



Output

LASCO/CACTUS Catalogs

Date		Characteristics					
2014/01/02	13:48:06	184	57	894	959	825	711
2014/01/03	00:24:05	264	18	225	272	169	0
2014/01/03	02:24:06	51	24	657	637	674	720
2014/01/03	03:47:08	61	44	1132	1303	961	965
2014/01/03	07:36:05	62	17	250	193	306	615
2014/01/03	10:36:05	65	21	316	273	358	627
2014/01/03	12:36:05	265	25	277	287	267	34
2014/01/03	18:00:06	154	60	208	114	295	430
2014/01/03	18:48:05	90	31	89	179	0	0
2014/01/03	19:36:05	222	112	286	331	237	0

Machine Learning Project

1st Part Data Enhancement

2nd Part CNN implementation

Improving Input Project

3rd Part Pre-processing Tool & History Maps

1st Part Data Reduction

2nd Part CNN implementation

3rd Part Pre-processing Tool & History Maps

The Data Reduction Project

Data cleaning (Multiple CME) + Data enhancement (Halo CME) + Data integration (No CME dates) = Enhanced CACTUS/LASCO catalogs

~~0025,1997/09/21 02:07,01,244,014,0484,0020~~
0026,1997/09/21 02:07,01,288,018,0384,0027

$\theta > 90^{\circ}$

0006,1997/09/09 20:06,03,279,108,0609,0124,0316,0813, ✓
0007,1997/09/13 06:25,01,258,014,0349,0771,0237,1922, ✗

Add non-CME date when:

$$t_{i+1} - t_i \geq 3\ [h]$$

Date	CME [0/1]	Halo CME [0/1]
2014/01/01 00:12:05	1	0
2014/01/04 23:12:05	1	1
2014/05/03 20:42:00	0	0

1st Part Data Enhancement

2nd Part CNN implementation

3rd Part Pre-processing Tool & History Maps

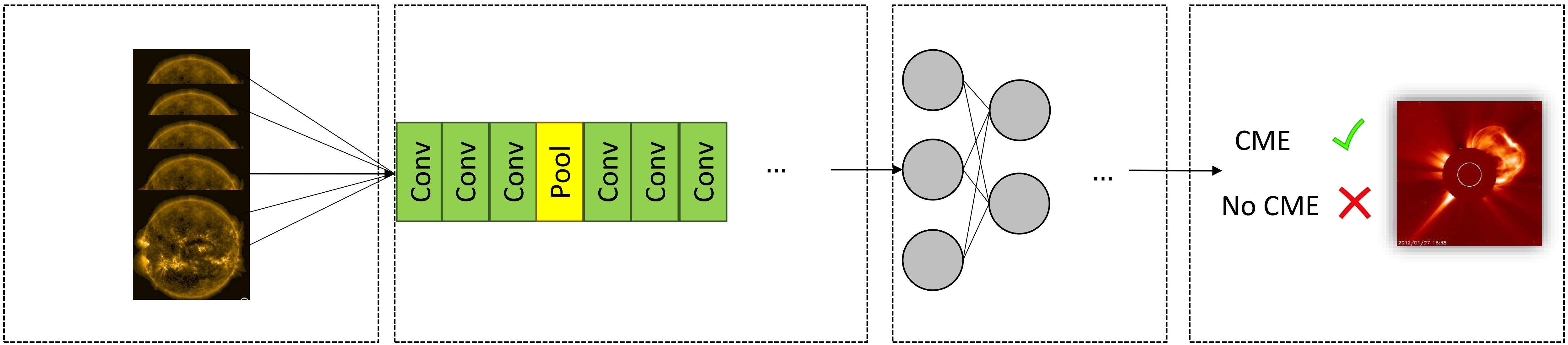
The Machine Learning Project

Input
[13,512,512]

Convolution Neural Network
Feature Extraction

Dense Neural Network
Classification

Output
CME / No CME

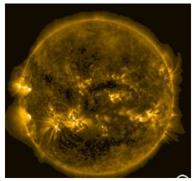


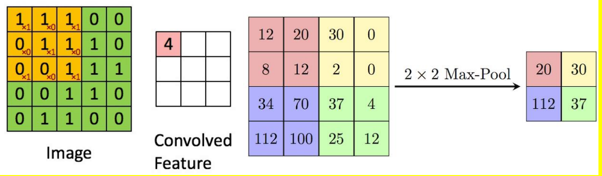
Input = 13 SDO images, 2 [h] history before the event.

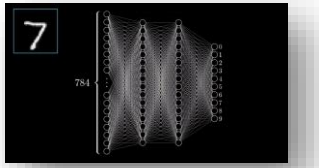
Output = 1/0

Final CNN architecture

Layer	Details & Operations	Output shape
Input	-	[512,512,13]
Convolution	Convolution [14] & 3x3 Kernel	[510,510,14]
Convolution	Convolution [16] & 3x3 Kernel	[508,508,16]
Convolution	Convolution [18] & 3x3 Kernel	[506,506,18]
Max Pooling	Max Pooling with 2x2 Kernel	[253,253,18]
Dropout	20 % Dropout	[253,253,18]
Convolution	Convolution [20] & 3x3 Kernel	[251,251,20]
Convolution	Convolution [28] & 3x3 Kernel	[249,249,28]
Convolution	Convolution [36] & 3x3 Kernel	[247,247,36]
Max Pooling	Max Pooling with 2x2 Kernel	[247,247,36]
Dropout	20 % Dropout	[123,123,36]
Convolution	Convolution [40] & 3x3 Kernel	[121,121,40]
Convolution	Convolution [56] & 3x3 Kernel	[119,119,56]
Convolution	Convolution [72] & 3x3 Kernel	[117,117,72]
Max Pooling	Max Pooling with 2x2 Kernel	[58,58,72]
Dropout	40 % Dropout	[253,253,18]
Convolution	Convolution [80] & 3x3 Kernel	[56,56,80]
Convolution	Convolution [112] & 3x3 Kernel	[54,54,112]
Convolution	Convolution [144] & 3x3 Kernel	[52,52,144]
Max Pooling	Max Pooling with 2x2 Kernel	[26,26,144]
Flatten	Flattening of the input	97344
Fully Connected	400 Neuron - Dense layer	400
Fully Connected	200 Neuron - Dense layer	200
Fully Connected	2 Neuron - Dense layer	2
Output	Classifier, 0.5 Threshold Sigmoid	2



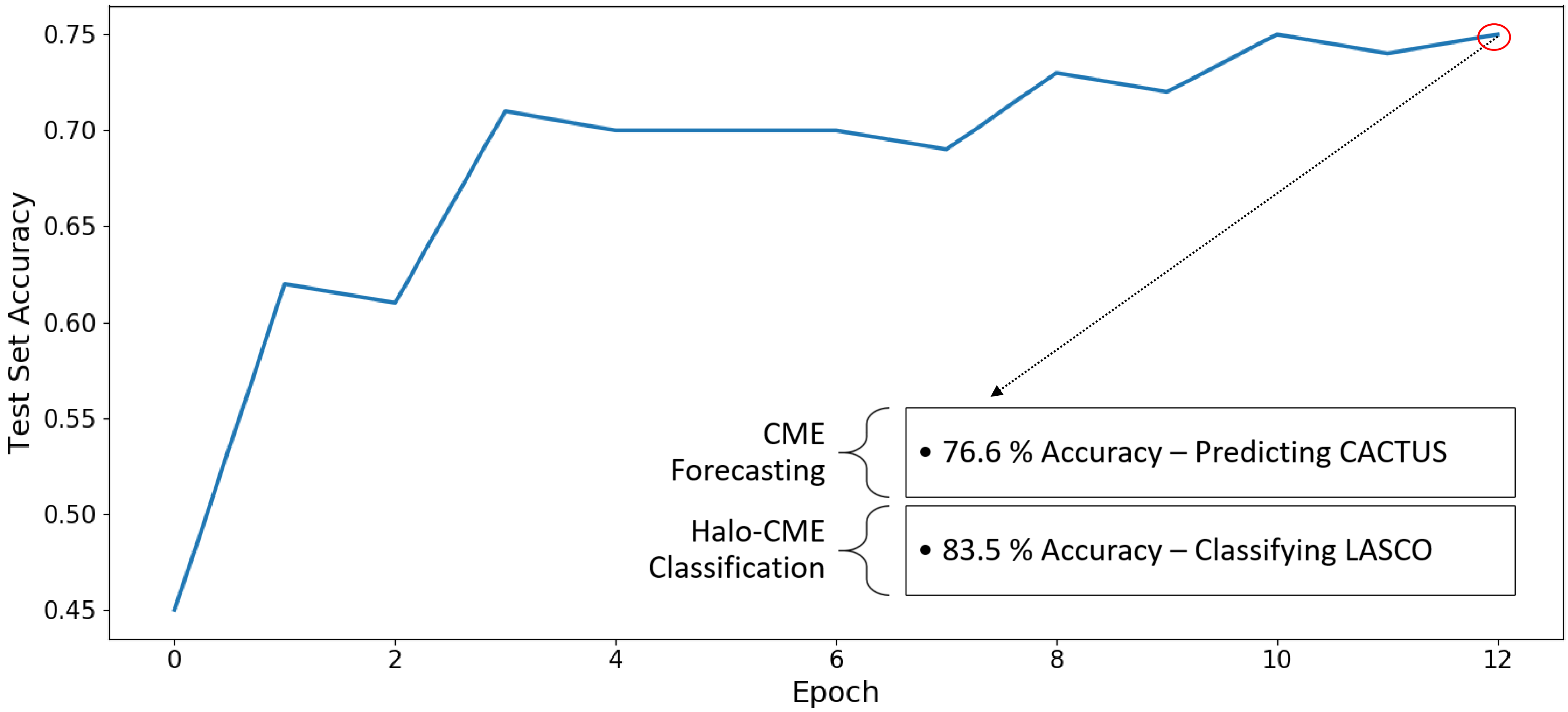




CME

No CME

Result of CNN



1st Part Data Enhancement

2nd Part CNN implementation

3rd Part Pre-processing Tool & History Maps

Pre-processing Tool – Motivation

Previous input

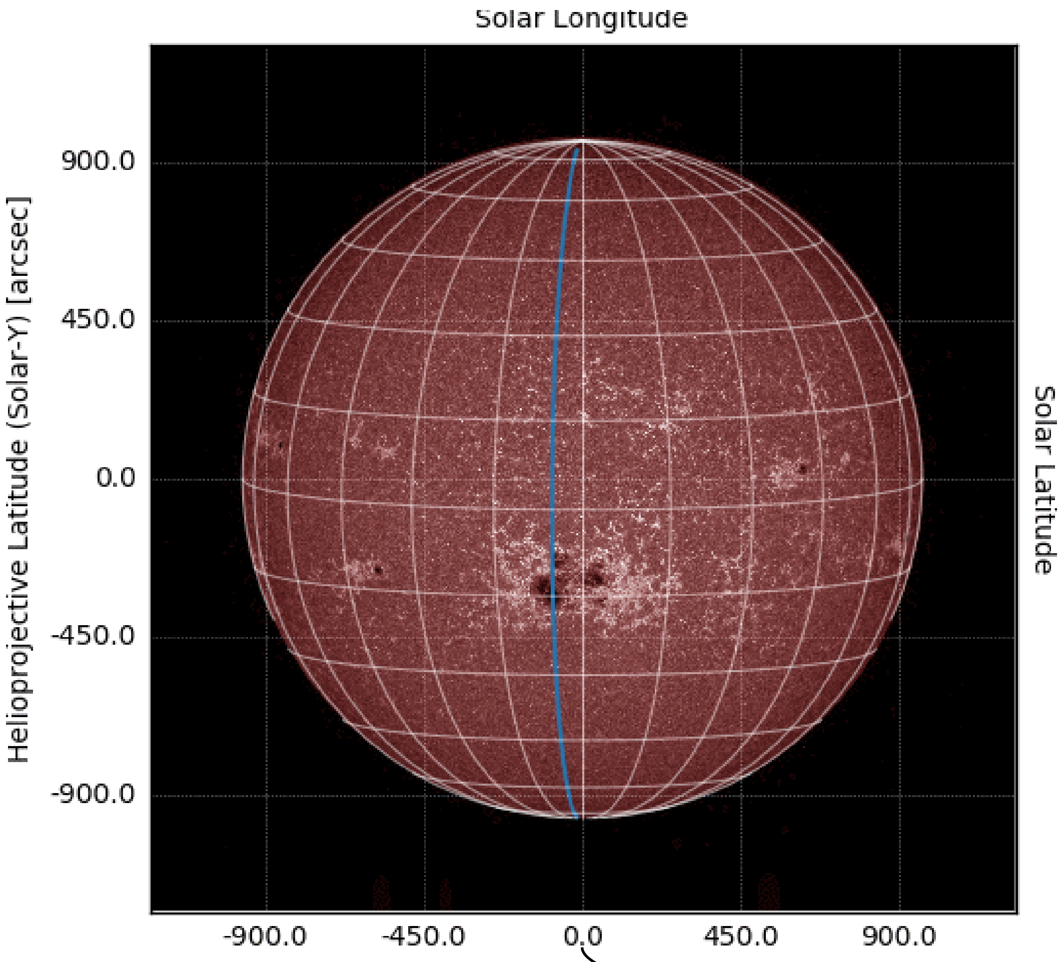
(+) Promising results.

(-) Expensive computationally and memory wise.

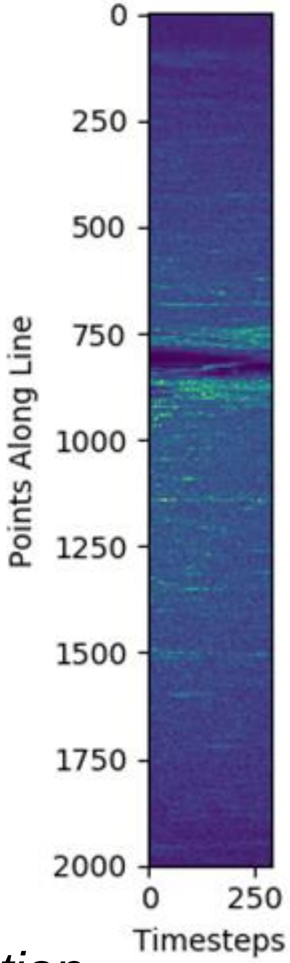
Using the pre-processing tool

(+) **New input** → less computational time & memory consumption.
(?) Better results.

History Map (HM) – Single Line Example

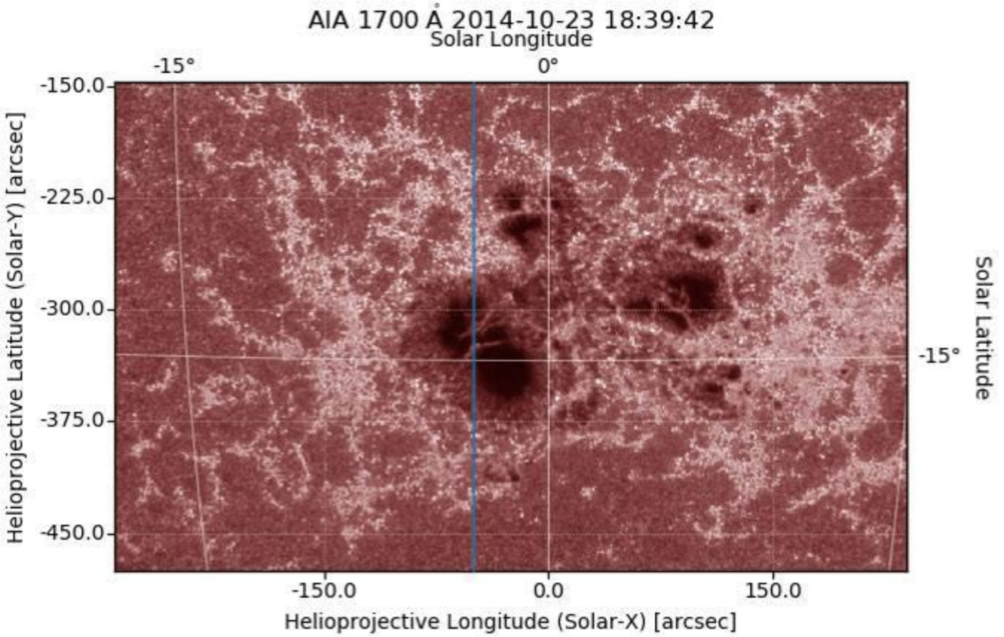


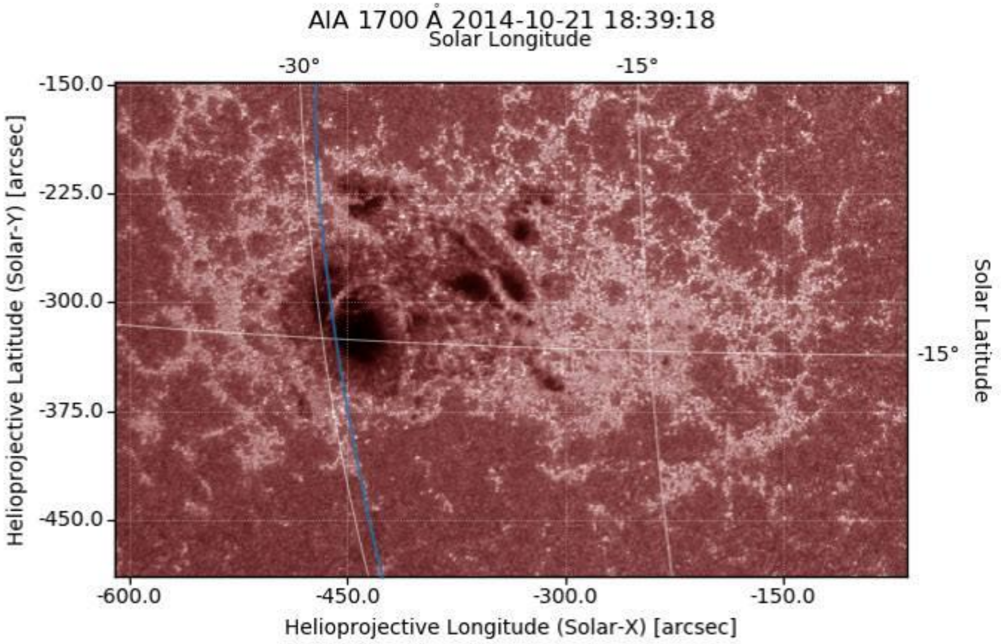
HM

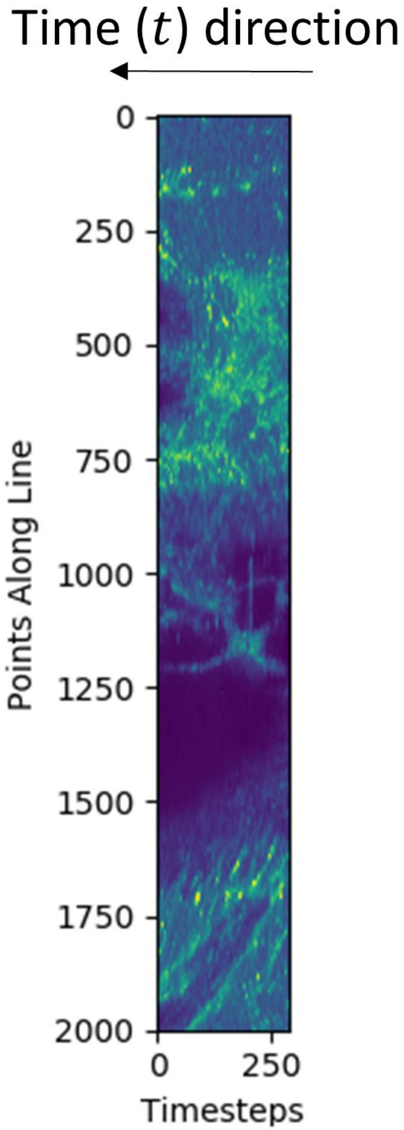


Same blue line, different representation

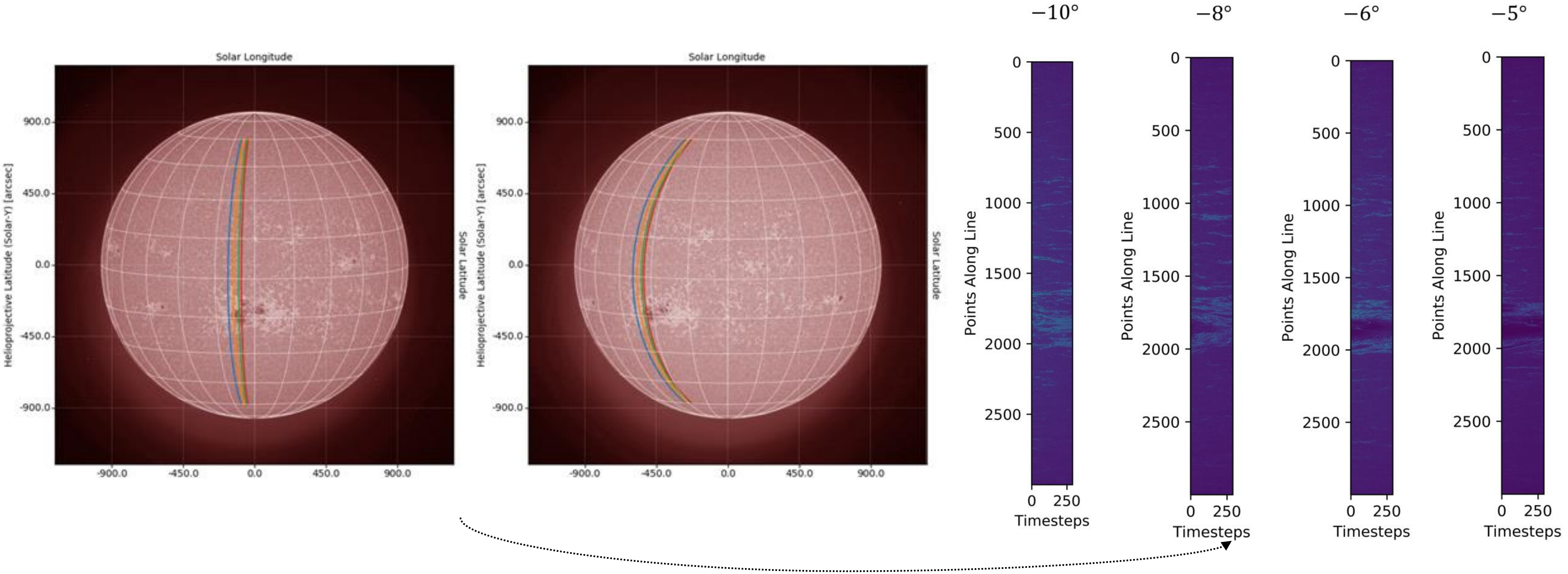
Pre-processing Tool – Sunspot







History Map – Multi Line Example

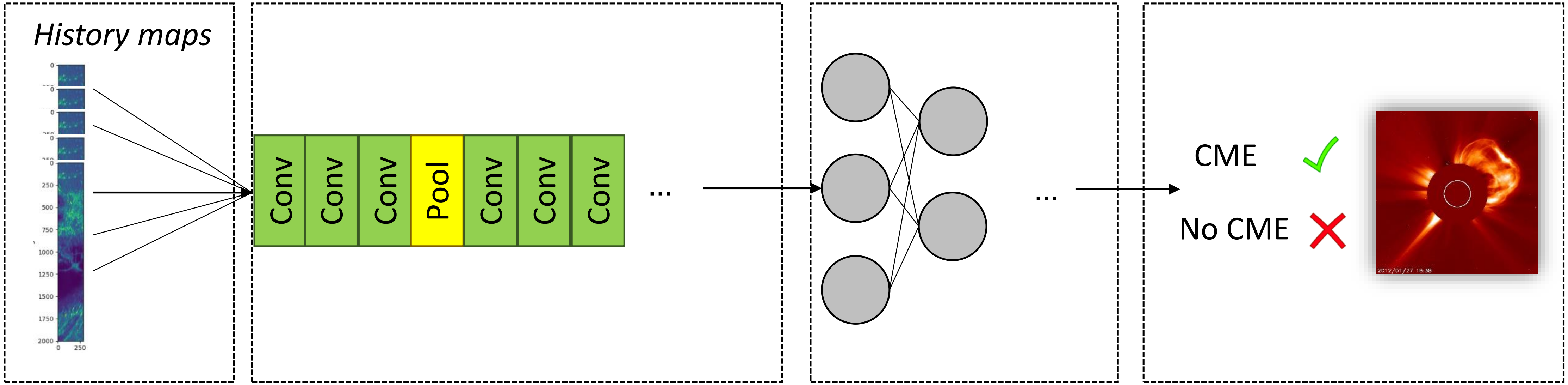


Why History Maps ?

- Substantial decrease of data and computational time.
- Structures are shown in a frame that is co-moving → time evolution is shown.
- Possibly useful for forecasting other phenomena such as Solar flares or Sunspots (Adaptable time scale).

Possible Future Project (?)

Input	Convolution Neural Network	Dense Neural Network	Output
$[C, x, y]$	Feature Extraction	Classification	CME / No CME



Conclusion

Summary

1. Enhanced, clean and processed SDO data and CACTUS/LASCO catalogs

2. Created multiple CNN models, with the best obtaining 76.6% prediction on CMEs and 83.5% classification between CME and halo-CME.

3. Created a pre-processing tool that derives “History Maps” (HM). Possibly useful in future Machine learning research and Solar data analysis.

Machine Learning & A.I. in Solar & Space

- An active community, many PhD and Postdoc positions currently around Europe (Italy, Belgium) – AIDA Project (https://aida-space.blogspot.com/)

- New specialized conferences (https://event.cwi.nl/ml-helio-2019/)

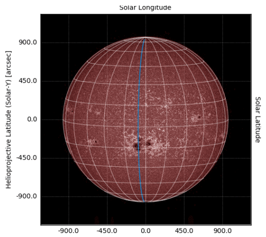
  Machine Learning in Heliophysics

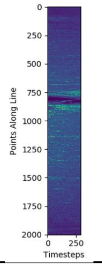
  16 - 20 September 2019 – Amsterdam

- New internship programs Nasa Frontier Lab (https://frontierdevelopmentlab.org/) for PhDs and Postdocs

Extras

Pre-processing Tool – Procedure & Output





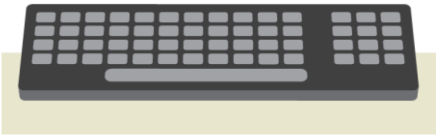
Procedure

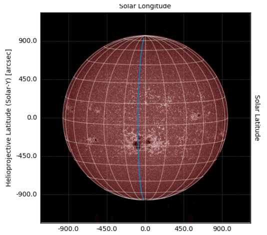
- Download data
- Track Sun’s differential rotation for every longitude line
- Go to next date on Catalog
- Repeat

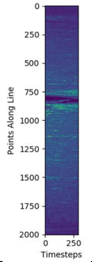
Output

1) Database file (.sql)
2) Scientific images (.FITS)
3) History maps (.png)
4) NN input (.npy)
5) Animation (.gif)

Pre-processing Tool – Design

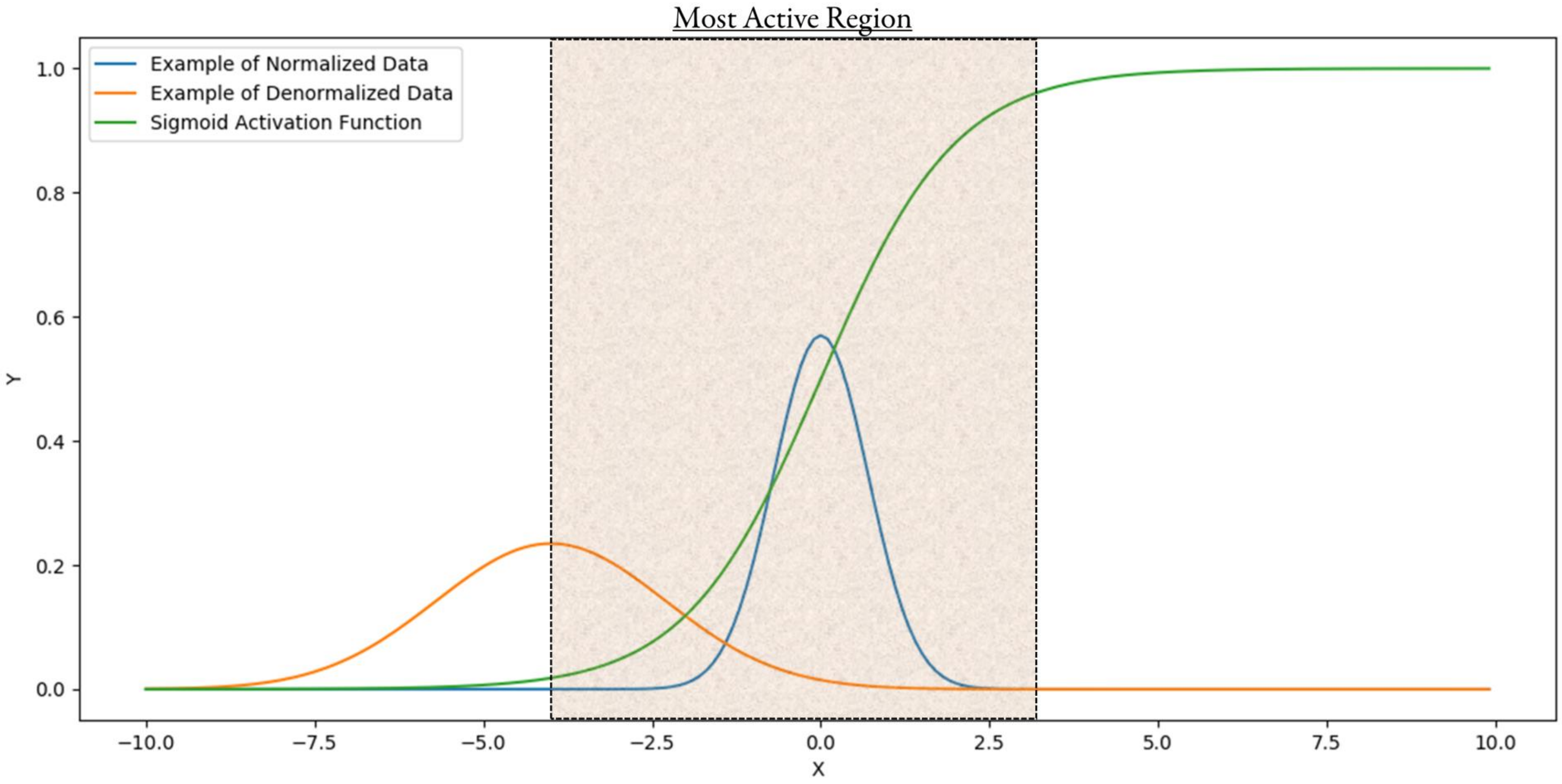




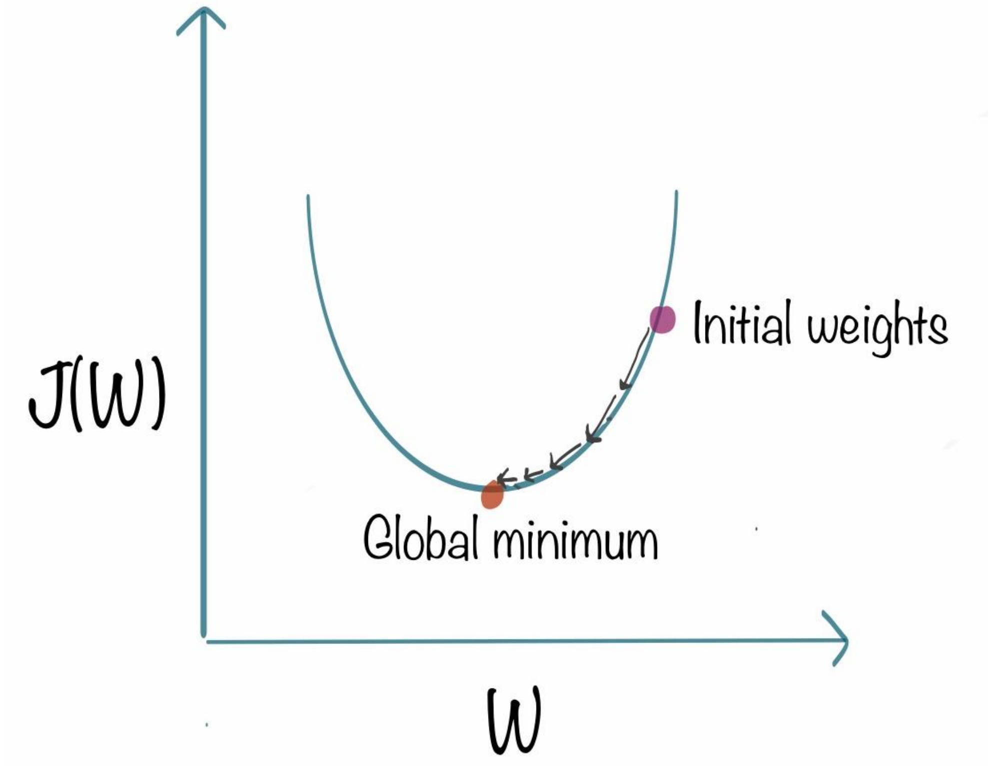


Input	Procedure	Output
1) Date – Date of event	• Download data for T [h] before event	1) Database file (.sql)
2) x – Points on line	• Track Sun's differential rotation for every line (y) using dt step	2) Scientific images (.FITS)
3) y – Longitude lines	• Go to next $date$ on Catalog (c)	3) History maps (.png)
4) dt – Time-step	• Repeat	4) NN input (.npy)
5) T – Total time		5) Animation (.gif)
6) λ – Wavelength		
7) C – Catalog		

Why normalization is vital?



Gradient Descent - Training



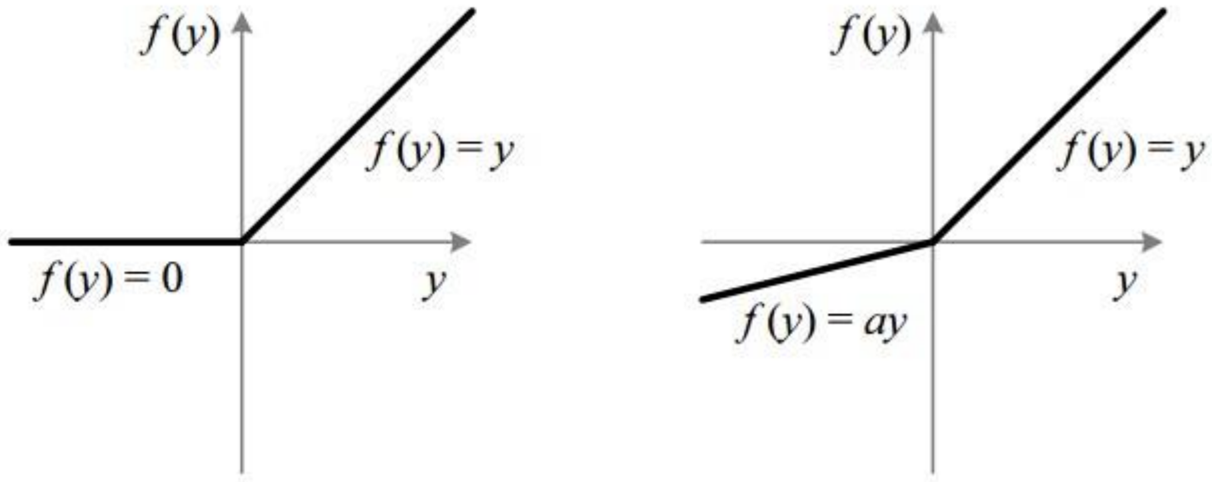
Loss Function/Error:

$$E = \frac{1}{2}\sum_{i}(a_i - t_i)^2$$

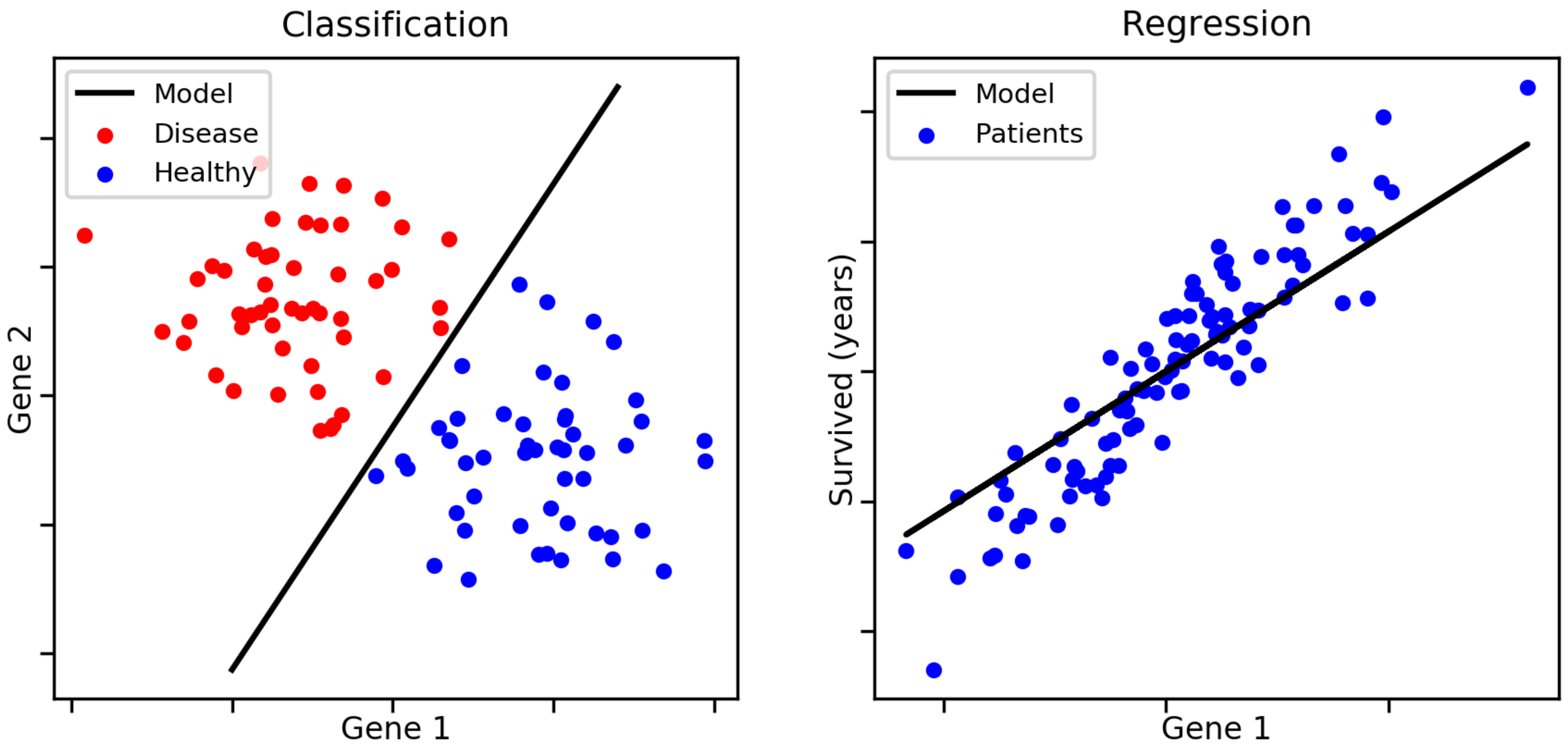
Advanced Activation functions

Goal → Complexity

Non-linear activations (Hidden Layers)

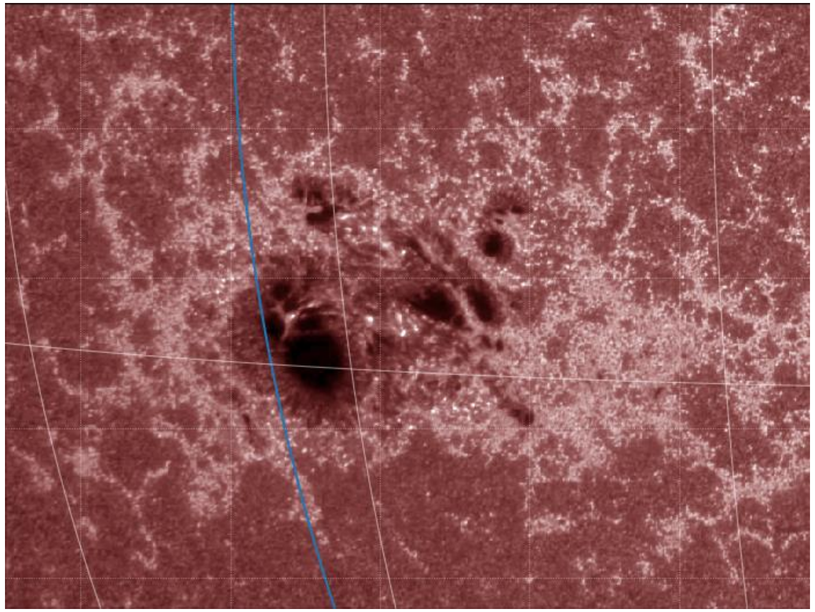


Type of Machine Learning Problems

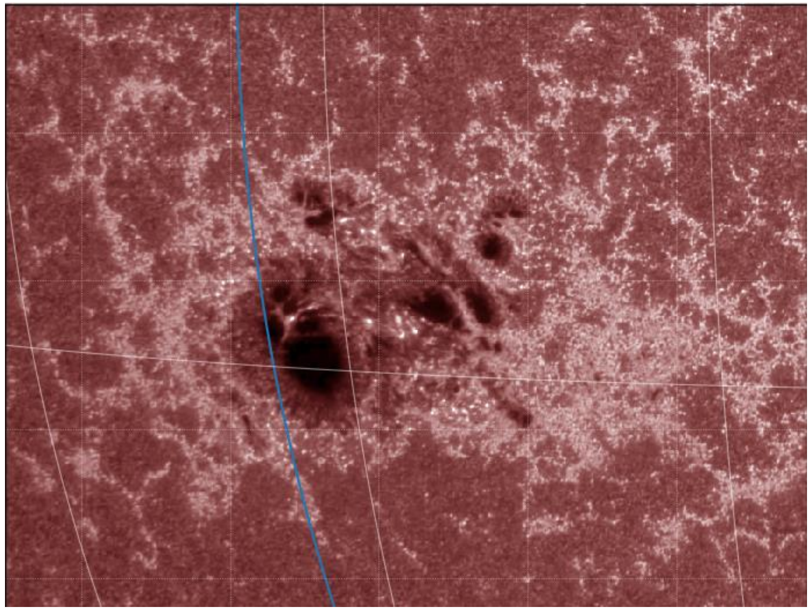


Differential Rotation Models

Allen



Howard



Snodgrass